(19) RÉPUBLIQUE FRANÇAISE

INSTITUT NATIONAL
DE LA PROPRIÉTÉ INDUSTRIELLE

PARIS

(11) **N° de publication :** **2 765 362**
(à n'utiliser que pour les commandes de reproduction)

(21) **N° d'enregistrement national :** **97 07996**

(51) Int Cl^6 : **G 06 F 12/02,** G 06 F 12/14

(12) DEMANDE DE BREVET D'INVENTION A1

(22) **Date de dépôt :** 26.06.97.

(30) **Priorité :**

(43) **Date de mise à la disposition du public de la demande :** 31.12.98 Bulletin 98/53.

(56) **Liste des documents cités dans le rapport de recherche préliminaire :** *Se reporter à la fin du présent fascicule*

(60) **Références à d'autres documents nationaux apparentés :**

(71) **Demandeur(s) :** *BULL CP8 SOCIETE ANONYME* — FR.

(72) **Inventeur(s) :** AJDENBAUM JEROME, HAMEAU PATRICE et PRESA ANNE FRANCE.

(73) **Titulaire(s) :**

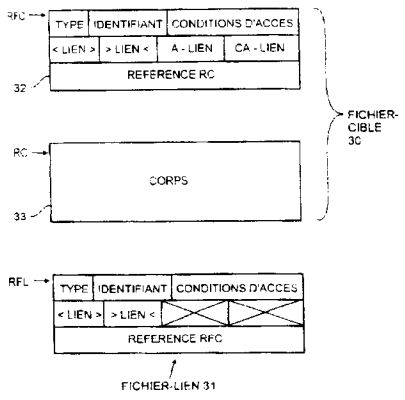
(74) **Mandataire(s) :** BULL SA.

(54) MODULE DE SECURITE COMPORTANT DES MOYENS DE CREATION DE LIENS ENTRE DES FICHIERS PRINCIPAUX ET DES FICHIERS AUXILIAIRES.

(57) L'invention concerne un module de sécurité coopérant avec un dispositif de traitement de l'information et comportant des moyens de traitement de l'information et des moyens de stockage de l'information, ces derniers stockant plusieurs fichiers.

Selon l'invention, le module de sécurité comprend:

- des moyens de création de lien agencés pour créer un lien entre au moins un fichier principal (30) et un fichier auxiliaire (31), le fichier principal ayant un contenu déterminé (33) et étant rendu accessible aux moyens de traitement dans les moyens de stockage grâce à des données de localisation (RFC), les moyens de création de lien associant au fichier auxiliaire (31) lesdites données de localisation; et
- des moyens de branchement agencés pour mettre à disposition des moyens de traitement, lorsque ceux-ci exécutent une demande d'accès visant à accéder au fichier auxiliaire (31), ledit contenu (33) du fichier principal (30) en utilisant lesdites données de localisation (RFC).



FR 2 765 362 - A1

Module de sécurité comportant des moyens de création de liens entre des fichiers principaux et des fichiers auxiliaires

L'invention est relative à un module de sécurité agencé pour coopérer avec un dispositif de traitement de l'information et comportant des moyens de traitement de l'information et des moyens de stockage de l'information, les moyens de stockage stockant plusieurs fichiers.

Le terme "module de sécurité" doit être pris, soit dans son sens classique dans lequel il désigne un dispositif ayant vocation, dans un réseau de communication ou d'information, à être détenu par un organisme supervisant le réseau et à stocker de façon protégée des paramètres secrets et fondamentaux du réseau tels que des clés cryptographiques, soit comme désignant plus simplement un dispositif attribué à divers usagers du réseau et permettant à chacun d'eux d'avoir accès à celui-ci, ce dernier dispositif étant lui aussi susceptible de détenir des paramètres secrets. Le module de sécurité pourra prendre la forme d'un objet portatif du type carte à puce.

La présente invention concerne notamment les cartes à micro-circuit et, plus généralement les objets portatifs dotés de circuits intégrés comportant au moins un microprocesseur, une mémoire morte (ROM) contenant un système d'exploitation de la carte et une ou plusieurs mémoires non volatiles, programmables par le microprocesseur. Ces mémoires non volatiles permettent de stocker des données et du code.Le microprocesseur contrôle le transfert des informations et, le cas échéant mémorise les données reçues de l'extérieur ou les lit pour les transmettre à l'extérieur. Ces objets possèdent un ou plusieurs moyens de communication. Les mémoires peuvent être de technologie EPROM, EEPROM, FeRAM, SRAM ou FLASH.

Grâce à l'évolution de la technologie, la taille de la mémoire ROM contenant le programme est de plus en plus importante ; ainsi les concepteurs de ce programme peuvent introduire de plus en plus de fonctions. Certaines de ces fonctions se rapportent directement à l'organisation de la mémoire programmable. Cette mémoire, dont la taille a aussi augmenté, est organisée hiérarchiquement en fichiers indépendants, cette organisation est décrite notamment dans la norme ISO 7816-4.

De nos jours, les cartes peuvent servir à de multiples applications et pour cela elles possèdent souvent deux ou trois niveaux hiérarchiques appelés par exemple : CARTE, APPLICATION et SERVICE. A chaque niveau, les informations se rapportant à une même affectation sont regroupées en fichiers, ces fichiers comprenant deux niveaux de stockage , à savoir des "REPERTOIRES" et, dans chaque répertoire, des informations de même nature stockées dans des "FICHIERS DE DONNEES ".

Cette architecture définie en plusieurs niveaux s'élabore généralement lors de la personnalisation de la carte, c'est-à-dire avant son utilisation. Il est possible cependant de rajouter en utilisation d'autres répertoires ou d'autres fichiers de données , mais cela dépend de la place disponible restant dans la mémoire non volatile programmable. Cette mémoire étant de taille limitée, il est important de ne pas gaspiller d'emplacement et de définir lors de la personnalisation uniquement la place nécessaire et suffisante au bon fonctionnement des répertoires et des fichiers de données .

Un excellent moyen de ne pas perdre de place consiste à ne pas dupliquer les informations. Ainsi, il faut éviter que les mêmes informations utiles à plusieurs répertoires soient écrites de façon identique dans plusieurs endroits de la mémoire. Malheureusement, l'architecture hiérarchisée des répertoires empêche de partager un même fichier de données entre plusieurs répertoires. Si deux répertoires doivent posséder les mêmes informations, il n'existe à ce jour que la solution de créer deux fichiers de données comportant ces mêmes informations à l'intérieur. La présente invention résout ce problème en évitant la duplication d'informations communes, tout en conservant les liens hiérarchisés entre fichiers de données et répertoires.

Un autre problème est celui de la mise à jour de fichiers présents en plusieurs endroits de la mémoire : il faut en effet effectuer les modifications nécessaires en chaque endroit. Outre le fait que cette opération est longue, elle peut être perturbée par une interruption accidentelle du fonctionnement de la carte , avec pour conséquence probable une mise à jour incomplète de l'ensemble de ces fichiers , c'est-à-dire une mise à jour de certains fichiers, et pas des autres. La cohérence entre tous ces fichiers ne serait plus assurée.

Par ailleurs, la structure en plusieurs niveaux peut pénaliser les temps d'accès à des fichiers de données ou répertoires de bas niveaux. En effet, pour

atteindre des données d'un répertoire d'un niveau inférieur, il faut dans de nombreux cas sélectionner tous les répertoires principaux de niveau supérieur. Par exemple, pour passer d'un répertoire à un autre de même niveau, il faut remonter une arborescence jusqu'à un premier répertoire commun puis redescendre, ceci en sélectionnant des répertoires intermédiaires. Ce mécanisme de sélection successive est lourd et pénalisant en temps.

La présente invention vise à résoudre ces différents problèmes : elle procure un moyen d'éviter la duplication en mémoire de données identiques ; elle assure la cohérence d'informations partagées entre plusieurs fichiers ; enfin, elle optimise la recherche d'informations dans des répertoires distants, dans l'arborescence des fichiers de la mémoire.

Elle concerne à cet effet un module de sécurité du genre cité au début de l'exposé, qui comprend :

-des moyens de création de lien agencés pour créer un lien entre au moins un fichier principal et un fichier auxiliaire, le fichier principal ayant un contenu déterminé et étant rendu accessible aux moyens de traitement dans les moyens de stockage grâce à des données de localisation, les moyens de création de lien associant au fichier auxiliaire lesdites données de localisation ;

-des moyens de branchement agencés pour mettre à disposition des moyens de traitement, lorsque ceux-ci exécutent une demande d'accès visant à accéder au fichier auxiliaire , ledit contenu du fichier principal en utilisant lesdites données de localisation.

D'autres détails et avantages de la présente invention apparaîtront au cours de la description suivante d'une forme de réalisation préféré mais non limitative, au regard des dessins annexés sur lesquels :

La figure 1 présente une arborescence de plusieurs niveaux hiérarchiques dans une carte ;

La figure 2 présente une organisation typique de répertoires et de fichiers de données dans une carte ;

La figure 3 présente la structure détaillée de deux catégories fondamentales de fichiers utilisés dans l'invention ;

La figure 4 est un organigramme détaillant les étapes d'une procédure de création de fichier selon l'invention ; et

La figure 5 est le schéma d'un module de sécurité auquel est destinée l'invention , coopérant avec un dispositif de traitement de l'information.

Le dispositif de traitement de l'information 51 représenté sur la figure 5 comprend de façon connue en soi un microprocesseur 52 auquel sont reliés une mémoire ROM 53, et une mémoire RAM 54, des moyens 55 pour coopérer, avec ou sans contact physique, avec un module de sécurité 58, et une interface de transmission 57 permettant au dispositif de traitement de l'information de communiquer avec un autre dispositif semblable, soit directement, soit au travers d'un réseau de communication.

Le dispositif 51 peut en outre être équipé de moyens de stockage tels que des disquettes ou disques amovibles ou non, de moyens de saisie (tels qu'un clavier et/ou un dispositif de pointage du type souris) et de moyens d'affichage, ces différents moyens n'étant pas représentés sur la figure 5.

Le dispositif de traitement de l'information peut être constitué par tout appareil informatique installé sur un site privé ou public et apte à fournir des moyens de gestion de l'information ou de délivrance de divers biens ou services, cet appareil étant installé à demeure ou portable. Il peut notamment s'agir aussi d'un appareil de télécommunications.

Par ailleurs, le module de sécurité 58 inclut des moyens de traitement de l'information 59, une mémoire non volatile 60, une mémoire volatile de travail RAM 64, et des moyens 63 pour coopérer avec le dispositif de traitement de l'information. Ce module est agencé pour définir, dans la mémoire 60, une zone secrète 61 dans laquelle des informations une fois enregistrées, sont inaccessibles depuis l'extérieur du module mais seulement accessibles aux moyens de traitement 59, et une zone libre 62 qui est accessible depuis l'extérieur du module pour une lecture et/ou une écriture d'informations. Chaque zone de la mémoire non volatile 60 peut comprendre une partie non modifiable ROM et une partie modifiable EPROM, EEPROM, ou constituée de mémoire RAM du type "flash", c'est-à-dire présentant les caractéristiques d'une mémoire EEPROM avec en outre des temps d'accès identiques à ceux d'une RAM classique.

En tant que module de sécurité 58, on pourra notamment utiliser un microprocesseur à mémoire non volatile autoprogrammable, tel que décrit dans le brevet américain n° 4.382.279 au nom de la Demanderesse. Comme indiqué en

colonne 1, lignes 13-25 de ce brevet, le caractère autoprogrammable de la mémoire correspond à la possibilité pour un programme fi situé dans cette mémoire, de modifier un autre programme fj situé également dans cette mémoire en un programme gj. Bien que les moyens à mettre en oeuvre pour réaliser cette autoprogrammation puissent varier selon la technique utilisée pour concevoir les moyens de traitement de l'information 59, on rappelle que, dans le cas où ces moyens de traitement sont constitués par un microprocesseur associé à une mémoire non volatile et selon le brevet précité, ces moyens peuvent inclure :

-des mémoires tampon de données et d'adresses, associées à la mémoire ;

-un programme d'écriture dans la mémoire, chargé dans celle-ci et contenant notamment les instructions permettant le maintien d'une part de la tension de programmation de la mémoire, et d'autre part des données à écrire et de leurs adresses, pendant un temps suffisant, ce programme d'écriture pouvant toutefois être remplacé par un automate d'écriture à circuits logiques.

Dans une variante, le microprocesseur du module de sécurité 58 est remplacé -ou tout du moins complété- par des circuits logiques implantés dans une puce à semi-conducteurs. En effet, de tels circuits sont aptes à effectuer des calculs, notamment d'authentification et de signature, grâce à de l'électronique câblée, et non microprogrammée. Ils peuvent notamment être de type ASIC (de l'anglais « Application Specific Integrated Circuit »). A titre d'exemple, on peut citer le composant de la société SIEMENS commercialisé sous la référence SLE 4436 et celui de la société SGS-THOMSON commercialisé sous la référence ST 1335.

Avantageusement, le module de sécurité 58 sera conçu sous forme monolithique sur une seule puce.

En variante au microprocesseur à mémoire non volatile autoprogrammable décrit ci-dessus, le caractère sécuritaire du module de sécurité pourra résulter de sa localisation dans une enceinte inviolable.

La mémoire non volatile des cartes est organisée en fichiers qui peuvent être, comme rappelé précédemment, de deux types : répertoire ou fichier élémentaire de données. Chaque fichier élémentaire comprend un en-tête et un corps contenant des informations. Le niveau de hiérarchisation est précisé dans l'en-tête, on y trouve également les références du fichier, l'état ou phase de vie de

la carte, les conditions d'accès et la taille. En règle générale, l'en-tête contient l'ensemble des informations qui permettent de gérer les informations stockées dans le corps. Deux ou trois niveaux sont actuellement utilisés. En référence à la figure 1, et en général, le niveau supérieur est appelé "CARTE", et les niveaux inférieurs "APPLICATION" ou "SERVICE". On peut parfaitement envisager des cartes avec plus de trois niveaux ; dans l'exemple cité, trois niveaux sont décrits.

A titre d'exemple, une même carte peut être utilisée pour diverses applications telles que : la banque, la municipalité, le dossier médical, le radiotéléphone cellulaire, qui sont représentées par des répertoires de niveau APPLICATION. Dans l'application municipale, on trouve des parties telles que les transport publics, l'accès à la piscine et à la bibliothèque, le paiement du stationnement, qui sont représentées par des répertoires de niveau SERVICE.

La figure 2 illustre un exemple des liens hiérarchisés entre des fichiers dans la mémoire programmable d'une carte. Le répertoire CARTE contient deux répertoires APPLICATION 1 et 2 et le fichier élémentaire C1. Le répertoire APPLICATION 1 contient deux répertoires SERVICE A1-S1 et A1-S2 et le fichier élémentaire A1-1. Le répertoire SERVICE A1-S2 possède un seul fichier élémentaire de données : A1S1-1. Le répertoire APPLICATION 2 possède deux répertoires SERVICE A2-S1 et A2-S2. Le répertoire SERVICE A2-S1 possède deux fichiers élémentaires de données : A2S1-1 et A2S1-2. Le répertoire SERVICE A2-S2 possède un fichier élémentaire de données : A2S2-1. Tous ces fichiers occupent une place non négligeable dans la mémoire limitée de la carte, il est donc important d'optimiser l'occupation mémoire et d'éviter de dupliquer les mêmes informations dans plusieurs emplacements différents.

Dans certains cas, les mêmes informations sont utilisées par deux répertoires différents. Par exemple, les coordonnées bancaires d'un individu porteur d'une carte : nom et adresse du porteur, nom et coordonnées de la banque, numéro de compte, information sur le crédit ...etc. peuvent être stockées dans un fichier élémentaire, inclus dans le répertoire correspondant à l'application bancaire, par exemple : le fichier élémentaire A1-1 dans le répertoire APPLICATION 1, décrit dans la figure 2.

La carte peut aussi servir de carte-ville ; cette application est gérée par le répertoire APPLICATION 2. Elle permet notamment de payer les transports en commun, d'accéder à la bibliothèque municipale et à certaines activités culturelles

payantes (théatre, cinéma...). Ces services sont gérés par les deux répertoires SERVICE A2-S1 et A2-S2, hiérarchiquement dépendants du répertoire APPLICATION 2. Lorsque la carte sert de moyen de paiement, pour payer par exemple les trajets effectués dans les transports en commun, l'argent est débité directement sur le compte bancaire dont les coordonnées sont précisées dans le fichier élémentaire A1-1. Il faut donc rendre accessible depuis le répertoire SERVICE A2-S1 du répertoire APPLICATION 2, les informations du fichier élémentaire A1-1 de APPLICATION 1. Cet accès est symbolisé par la flèche sur la figure 2. La solution consistant à reproduire les données n'est pas satisfaisante.

Un autre exemple concerne les clés secrètes et les codes confidentiels : leurs valeurs peuvent être identiques lors de l'accès à différentes répertoires qui ne sont pas hiérarchiquement dépendants. Le problème est important si des clés de type RSA (des inventeurs Rivest, Shamir et Adleman) stockées sur plus de 1024 bits sont utilisées. Un dernier exemple concerne les fichiers élémentaires de ratification : ces fichiers élémentaires servent à mémoriser les bonnes ou mauvaises présentations de clés ou codes. Le regroupement de plusieurs fichiers élémentaires de ratification correspondant à des clés différentes permet de gagner de la place et d'accroître la sécurité.

Une façon de réaliser l'invention consiste à créer et gérer des fichiers dits "Lien" dont le corps est confondu avec celui d'autres fichiers. L'invention consiste à pouvoir partager un même corps de fichier entre plusieurs fichiers. Ceci peut être réalisé en indiquant, soit dans l'en-tête du fichier, soit dans son corps, l'adresse où se situent effectivement les données.

Sur la figure 3, sont représentés deux fichiers, à savoir un fichier-cible 30 et un fichier-lien 31. La description qui suit concerne aussi bien le cas où ces fichiers sont des fichiers de données et celui où ils représentent des répertoires. Ces répertoires contiennent soit une arborescence de sous-répertoires donnant accès à des fichiers de données, soit des fichiers de données qui leur sont directement rattachés, soit les deux. Le terme « données » regroupe à la fois des données non exécutables et des données exécutables ou programmes.

Le fichier-cible 30 est organisé, dans cet exemple, en deux parties comprenant un en-tête 32 et un corps 33. L'en-tête 32 inclut un premier groupe de paramètres connus en eux-mêmes, à savoir :

-un type, qui indique si le fichier est un répertoire ou bien un fichier de données ;

-un identifiant qui désigne le fichier au sein d'un répertoire qui le contient ; il s'agit par exemple d'un nom ou d'un numéro ; et

-des conditions d'accès qui donnent une liste de droits d'accès à ce fichier déterminé, pour tout usager : elles précisent par exemple si le fichier est accessible ou non en lecture ou écriture ; de façon connue en soi, la délivrance de ces droits peut être subordonnée à la présentation de clés ou mots de passe.

L'en-tête 32 inclut un second groupe de paramètres qui sont spécifiques à l'invention, à savoir :

-un paramètre <Lien> qui peut prendre deux valeurs : soit la valeur 1, qui indique que ce fichier est un fichier-lien soit la valeur 0 qui indique qu'il n'est pas un fichier-lien ; ici, ce paramètre a la valeur 0 ;

-un paramètre >Lien< qui peut prendre deux valeurs : soit la valeur 1, qui indique que ce fichier est un fichier-cible, soit la valeur 0 qui indique qu'il n'est pas un fichier-cible ; ici, ce paramètre a la valeur 1 ;

-un paramètre A-Lien qui peut prendre deux valeurs : soit la valeur 1, qui indique que ce fichier peut être lié à un fichier-lien , soit la valeur 0 qui l'en empêche ; et

-un paramètre CA-Lien qui définit des conditions de création que l'usager devra respecter lorsqu'il voudra créer un lien entre ce fichier et un fichier-lien : elles pourront par exemple définir des clés ou mots de passe à présenter par l'usager.

L'en-tête 32 comporte enfin une référence RC indiquant au microprocesseur de la carte une valeur binaire d'une adresse mémoire RC à partir de laquelle est stocké le corps 33 précité.

Dans une variante, le corps 33 est stocké en mémoire immédiatement à la suite de l'en-tête 32, de sorte que la mention de la référence RC n'est pas nécessaire.

Par ailleurs, si le fichier cible 30 est du type « répertoire » , le corps 33 contient soit une arborescence de sous-répertoires donnant accès à des fichiers

de données, soit des fichiers de données qui lui sont directement rattachés, soit les deux ;

Si au contraire le fichier-cible 30 est du type « fichier de données », le corps 33 contient un ensemble de données directement accessibles pour lecture ou modification, ou exécutables par le microprocesseur de la carte .

En variante , l'organisation du fichier-cible 30 pourra être différente de celle en deux parties (en-tête et corps) présentée sur la figure 3. Ainsi par exemple , les paramètres de l'en-tête 32 pourront être répartis en des endroits spécifiques du corps.

Quant au fichier-lien 31, il ne comprend qu'une seule partie, à savoir un en-tête qui présente la même structure que celui 32 du fichier-cible 30, mais a un contenu qui en diffère de la façon suivante :

-s'agissant d'un fichier-lien , et non d'un fichier-cible, les paramètres A-Lien et CA-Lien ne sont en général pas utilisés, sauf dans le cas particulier décrit plus loin ;

-par ailleurs, la « référence » n'est pas celle relative à un éventuel corps rattaché au fichier-lien , mais une référence RFC précisant la localisation en mémoire d'un fichier-cible ainsi lié à ce fichier-lien. Dans cet exemple , c'est le fichier-cible 30. La référence RFC est soit de préférence « physique » et constituée par une valeur binaire d'une adresse mémoire à partir de laquelle est stocké le fichier-cible 32 précité, soit en variante « logique » et constituée par un chemin d'accès précisant les identifiants d'un ou plusieurs répertoires à partir desquels le fichier-cible 32 est accessible.

Dans le cas très particulier d'une gestion dynamique de la mémoire de la carte, celle-ci peut être ré-organisée par le microprocesseur afin d'optimiser l'occupation de la mémoire. L'emplacement des fichiers peut donc fluctuer, ainsi par conséquent que leurs adresses d'implantation. Dans ce cas, seule la référence logique du fichier-cible est facilement utilisable car les adresses physiques risquent de constamment changer. En prenant comme exemple le cas évoqué précédemment, la référence logique est : [CARTE → APPLICATION 1 → Fichier de données A1-1].

Il apparaît donc qu'un fichier-lien est dépourvu de corps, mais est lié à un fichier-cible déterminé, dont le corps sera ainsi mis à disposition du fichier-lien.

On notera que, dans un cas particulier, un second fichier-lien, différent du fichier-lien 31, pourrait être lié, non pas directement à un fichier-cible , mais par exemple au fichier-lien 31. La situation serait alors la suivante :

-la référence contenue dans le second fichier-lien serait celle du fichier-lien 31 ;

-le paramètre CA-Lien serait avantageusement utilisé dans l'en-tête du premier fichier-lien pour contrôler les conditions de création du second fichier-lien .

En fonctionnement, lors de la sélection d'un fichier par l'usager , un programme du microprocesseur lit son en-tête et teste son paramètre <Lien>. S'il est égal à 0, le fonctionnement est conforme à l'art antérieur : le corps estdirectement rattaché à cet en-tête.

Si <Lien> est égal à 1, le fichier est un fichier-lien. Le programme lit en conséquence la référence RFC de ce fichier-lien précisant l'adresse ou le chemin d'accès d'un fichier-cible contenant un corps indirectement rattaché à l'en-tête du fichier-lien. Avant de mettre à disposition de l'usager ou du microprocesseur le contenu du corps du fichier-cible , le programme effectue les vérifications suivantes, en consultant les en-têtes respectifs du fichier-lien et du fichier-cible :

-lors de la création du fichier-lien, il vérifie que le type du fichier-cible identifié est identique à celui du fichier-lien ; dans la négative, la procédure d'accès au fichier-lien est interrompue ;

-lors de chaque accès au fichier-cible, il vérifie le respect des conditions d'accès selon une procédure qui sera précisée plus loin.

Ces vérifications étant effectuées, le programme poursuit sa procédure d'accès au contenu du fichier-lien.Si la taille du corps du fichier-cible est « zéro octet » , c'est-à-dire s'il ne contient rien, le programme s'interrompt et la carte renvoie un message d'erreur. Sinon, le programme recherche les informations contenues dans ce corps, à partir de l'adresse RC.. En reprenant l'exemple de la figure 2, on suppose que le fichier A1-1 du répertoire de l'application 1 a été créé sous forme de fichier-cible, et que les fichiers A2S1-1, A2S1-2, A2S2-1 des répertoires service A2-S1 et A2-S2 ont été créés sous forme de fichiers-lien. En conséquence, la sélection d'un fichier-lien tel que A2S1-1 donnera accès au contenu du fichier-cible A1-1.

Les conditions d'accès au corps du fichier-Cible, définies dans l'en-tête de celui-ci, doivent dans tous les cas être respectées, lors de l'exécution d'un lien entre fichier-lien et fichier-cible . Plusieurs stratégies sont possibles. La plus simple consiste à obéir aux conditions d'accès définies dans l'en-tête du fichier-Cible :ainsi l'accès des informations dans le fichier-Cible via le fichier-lien n'est accordé que si les conditions d'accès du fichier-Cible sont respectées.

Une autre stratégie consiste à prendre en compte les conditions d'accès du fichier-Cible lors de la création du fichier-lien. Il faut alors vérifier que les conditions d'accès inscrites dans l'en-tête du fichier-lien incluent toutes les conditions d'accès du fichier-Cible auquel il va être lié.

Une troisième stratégie est applicable lorsque les conditions d'accès s'expriment sous la forme d'une valeur binaire : elle consiste à cumuler les deux conditions d'accès. Concrètement, cette opération peut être réalisée en effectuant un ET logique entre les deux valeurs. L'accès des informations dans le fichier-Cible via le fichier-lien n'est accordé que si, à la fois, les conditions d'accès des fichiers-Cible et lien sont respectées.

A travers ces différentes stratégies, le lecteur comprend que l'objectif, au niveau du contrôle d'accès, consiste à proscrire toute possibilité de contourner les conditions d'accès d'un fichier en le liant à un fichier qui possède des conditions d'accès plus favorables. D'autres stratégies, connues de l'homme du métier, pourront être utilisées pour assurer la sécurité d'accès.

Un perfectionnement important concernant la sécurité consiste à utiliser le paramètre A-Lien d'un fichier-cible pour l'empêcher d'être lié à un autre fichier . Si la valeur de ce paramètre est "1", lors de la création d'un fichier-lien rattaché à ce fichier-cible , l'opération est menée à bien et le contenu du corps du fichier-Cible est bien accessible par le fichier-lien. Si en revanche la valeur de ce champ est "0", ce fichier-cible ne peut être lié à aucun autre. Lors d'une tentative de création d'un fichier-lien désignant un fichier dont le champ A-Lien égal à "0", l'opération est refusée et la carte rend un message d'erreur.

Un problème se pose lors de l'effacement, c'est-à-dire de la suppression de fichiers Lien/Cible. Si un fichier-Cible est effacé, l'accès par des fichiers-Liens aux informations qu'il contenait n'est plus possible. Les conséquences d'un tel

effacement ne sont donc pas uniquement limitées au répertoire qui contenait ce fichier, mais peuvent se répercuter dans d'autres répertoires. La solution consiste soit à interdire l'effacement du fichier-Cible auquel est rattaché un ou plusieurs fichiers-Lien , soit à avertir l'utilisateur de la carte que certains fichiers ne sont plus opérationnels après cet effacement. Un première méthode consiste donc à tester la valeur >Lien<. Si cette valeur est 1, le fichier que l'on est en train d'effacer est un fichier-Cible.L'opération est alors soit interdite , soit menée à bien mais avec un avertissement ; dans ce dernier cas, le terminal de commande de la carte doit effacer tous les fichiers-Lien liés au fichier-cible effacé.

Pour effacer un fichier-Cible sans perturber le reste de la mémoire, il faut donc préalablement effacer tous les fichiers-Lien qui lui sont rattachés. Lorsque le dernier fichier-Lien est effacé, l'indicateur >Lien< prend la valeur "0" : il ne s'agit donc plus d'un fichier-Cible. Un simple indicateur binaire n'est donc pas suffisant pour comptabiliser le nombre de fichiers-Lien sans devoir balayer chaque fois toute la mémoire non volatile de la carte . Une solution consiste à prévoir un compteur Cp-Lien à la place du paramètre >Lien<. Avantageusement, ce compteur est de 8 bits, ce qui autorise l'existence de 255 fichiers-Liens au maximum : ce nombre paraît largement suffisant pour des applications courantes. Ce compteur est incorporé dans l'en-tête du fichier-cible .

Lors de la création du fichier-cible, le compteur Cp-Lien est mis à "00". A chaque nouvelle création d'un fichier-Lien qui est rattaché au fichier-cible , le compteur est incrémenté. A chaque effacement d'un fichier-Lien qui lui est rattaché, le compteur est décrémenté. Avantageusement, lors de la sélection de ce fichier-cible, la valeur du compteur peut être émise avec les autres information d'en-tête : l'utilisateur peut ainsi connaître le nombre de fichiers-Lien rattachés au fichier-cible sélectionné.

Pour résoudre le problème de l'effacement d'un fichier-Cible de façon plus adroite, le programme de la carte est équipé d'une commande, activable de l'extérieur de la carte, permettant d'échanger les statuts respectifs « lien » et « cible » de deux fichiers. Ainsi, un fichier-Cible devenu un fichier-Lien peut être effacé sans conséquence pour les autres fichiers-Lien. Le contenu du nouveau fichier-Cible est alors constitué par celui de l'ancien fichier-cible . Cette opération est particulièrement facile lorsque les corps de fichiers sont physiquement séparés

des en-têtes. Pour des raisons de sécurité, l'exécution de cette commande est soumise à la vérification des conditions d'accès définies dans l'en-tête de l'ancien fichier-cible, et éventuellement à la vérification des conditions de création de liens entre fichiers, définies par le paramètre CA-Lien dans le répertoire du nouveau fichier-cible . Lors de l'exécution de cette commande d'échange, la valeur du compteur CP-Lien de l'ancien fichier-Cible est mémorisée dans le compteur CP-Lien du nouveau fichier-Cible.

La figure 4 illustre un procédé de création d'un fichier, qu'il s'agisse d'un fichier-lien ou non. Il inclut, outre des étapes spécifiques à l'invention et relatives au fichier-lien , certaines étapes connues en elles-mêmes et relatives à la création de tout fichier, quelle que soit sa nature. A l'étape 1, un ordre de création de fichier est reçu par la carte, accompagné de données de création : ces données définissent notamment le type et l'identifiant du fichier à créer et, s'il s'agit d'un fichier-lien , la référence RFC (figure 3) d'un fichier-cible auquel il doit être lié.

Ensuite, le système d'exploitation de la carte vérifie que la création d'un nouveau fichier est possible dans le répertoire actuel, appelé aussi "courant" (étape 2). En effet, la création d'un nouveau fichier est éventuellement soumise à la bonne présentation préalable de clés définies par les conditions d'accès de l'en-tête du répertoire courant. Puis, on vérifie qu'il reste suffisamment de mémoire dans le répertoire courant pour contenir le nouveau fichier (étape 3). Si un de ces tests est négatif, l'ordre de création est interrompu (étape 13), et la carte renvoie alors un message correspondant à l'origine de l'arrêt.

Le système d'exploitation teste ensuite s'il s'agit de la création d'un fichier normal ou de la création d'un fichier-Lien (étape 4). Une différence importante entre un fichier normal et un fichier-lien, et sur laquelle peut porter le test réside dans les données de création, et principalement dans l'indication précise de la localisation du fichier-Cible (adresse physique ou logique). Si ce n'est pas un fichier-Lien, le programme saute directement à l'étape 12 décrite ci-après. Dans le cas contraire, les informations correspondant au fichier-Cible désigné sont recherchées et analysées à l'étape 5 ; puis le système d'exploitation effectue un certain nombre de tests pour s'assurer que le lien entrele fichier-cible désigné et le fichier-lien à créer est possible.

Tout d'abord, on vérifie à l'aide des données de création que le fichier-Cible existe bien (étape 6). Si par contre, ces données ne correspondent à aucun fichier, l'opération de création est interrompue et la carte envoie un message d'erreur (étape 13). A l'étape 7, le paramètre A-Lien du fichier Cible localisé est testé. Si sa valeur est "1", l'opération peut être menée à bien. Sinon, le fichier-Cible ne peut être lié à aucun autre. L'opération de création est alors interrompue et la carte envoie un message d'erreur. A l'étape 8, le système d'exploitation teste si les éventuelles clés définies dans les conditions de création du fichier-Lien, c'est à dire définies par le paramètre CA-Lien du fichier-Cible, ont été préalablement présentées. Si ce n'est pas le cas, l'opération de création est interrompue.

A l'étape 9, le système d'exploitation de la carte vérifie que les types de fichiers et les conditions d'accès aux informations sont compatibles. Pour cela, le paramètre TYPE du fichier-Cible est comparé à celui transmis dans les données de création. Si les valeurs sont différentes, ou du moins incompatibles, comme par exemple dans le cas d'un ordre de création visant à faire un lien entre un fichier de données et un répertoire, ou un lien entre un fichier de données de type "public" et un fichier de données de type "secret", alors l'opération de création est interrompue et la carte envoie un message d'erreur. Ce test est facultatif car une autre solution consiste à forcer les données reçues pour le fichier-Lien à créer, à la même valeur que celles du fichier Cible désigné : la compatibilité est dans ce cas certaine.

Enfin, un dernier test effectué porte sur les conditions d'accès aux informations contenues dans le fichier-Cible (étape10). Il s'agit d'éviter de contourner les conditions d'accès du fichier-Cible par un fichier-Lien qui possèderait des conditions d'accès plus favorables. Une des stratégies décrites précédemment consiste à interdire la création d'un fichier-Lien possédant des conditions d'accès moins restrictives que celles du fichier-Cible : l'opération de création est alors interrompue et la carte envoie un message d'erreur (étape 13). Une autre stratégie consiste à aménager, et donc modifier, des conditions d'accès trop favorables du fichier-lien pour les rendre au moins aussi restrictives que celles du fichier-Cible. Dans ce cas, le test de l'étape 10 devient une opération de calcul avec modification, si besoin, des conditions d'accès transmises dans la commande.

Une fois les étapes de test franchies, la création du fichier-Lien peut intervenir. A l'étape 11, l'en-tête du fichier-Cible est mise à jour. Cela concerne principalement le paramètre >Lien< ou Cp-Lien. S'il s'agit du paramètre >Lien<, , le programme vérifie qu'il possède la valeur à "1", ou sinon le met à cette valeur. S'il s'agit au contraire du compteur Cp-Lien, , celui-ci est incrémenté d'une unité.

Enfin à l'étape 12, un nouveau fichier est effectivement créé, et les valeurs des paramètres d'en-tête de ce fichier sont déterminées en mémoire de travail à partir des données de création. Ces valeurs sont écrites en mémoire programmable non volatile. Si c'est un fichier-Lien qui est créé, une référence liée à la localisation du fichier-Cible (adresse physique ou logique) est écrite. Une fois toutes ces étapes franchies, la carte rend un message d'état correct et le fichier nouvellement créé est opérationnel.

Le cas de la création d'un fichier-Cible ne sera pas détaillé, puisque, lors de sa création, ce fichier est analogue à un fichier classique. Ce n'est qu'au moment où il est lié à un fichier-lien qu'il devient un fichier-cible effectif.

Une application particulièrement intéressante de l'invention et relative aux répertoires-lien est celle où un répertoire porte-monnaie électronique est utilisé par la carte pour permettre des paiements. Ce répertoire contient des fichiers élémentaires contenant des clés, des zones débit-crédits, des zones de validation de mot de passe, etc... Un tel répertoire peut être utilisé dans diverses applications (transport, restaurant, centrale d'achats) : chacune d'elles doit donc contenir un répertoire-lien lié au répertoire porte-monnaie électronique, lequel devient alors un répertoire-cible.

REVENDICATIONS

1. Module de sécurité agencé pour coopérer avec un dispositif de traitement de l'information et comportant des moyens de traitement de l'information et des moyens de stockage de l'information, les moyens de stockage stockant plusieurs fichiers , caractérisé en ce qu'il comprend :

-des moyens de création de lien agencés pour créer un lien entre au moins un fichier principal (30) et un fichier auxiliaire (31), le fichier principal ayant un contenu déterminé (33) et étant rendu accessible aux moyens de traitement dans les moyens de stockage grâce à des données de localisation (RFC), les moyens de création de lien associant au fichier auxiliaire (31) lesdites données de localisation ;

-des moyens de branchement agencés pour mettre à disposition des moyens de traitement, lorsque ceux-ci exécutent une demande d'accès visant à accéder au fichier auxiliaire (31), ledit contenu (33) du fichier principal (30) en utilisant lesdites données de localisation (RFC).

2. Module de sécurité selon la revendication 1, dans lequel le fichier auxiliaire (31) contient un paramètre (<lien>) indiquant qu'il contient des données de localisation d'un fichier principal (30).

3. Module de sécurité selon la revendication 1, dans lequel le fichier principal (30) contient un paramètre (>lien<) indiquant que ses données de localisation (RFC) sont associées à au moins un fichier auxiliaire (31).

4. Module de sécurité selon la revendication 3, dans lequel ledit paramètre (>lien<) est une valeur d'un compteur (Cp-Lien) agencé pour compter un nombre de fichiers auxiliaires.(31) qui sont liés à ce fichier principal (30).

5. Module de sécurité selon la revendication 1, dans lequel le fichier principal (30) contient un paramètre (A-Lien) indiquant si la mise à disposition de son contenu lors d'une demande d'accès au fichier auxiliaire (31) est autorisée ou non.

6. Module de sécurité selon la revendication 1, dans lequel le fichier principal (30) contient un paramètre (CA-Lien) définissant des conditions de création à respecter par les moyens de traitement lors de la création d'un lien avec un fichier auxiliaire (31) déterminé.

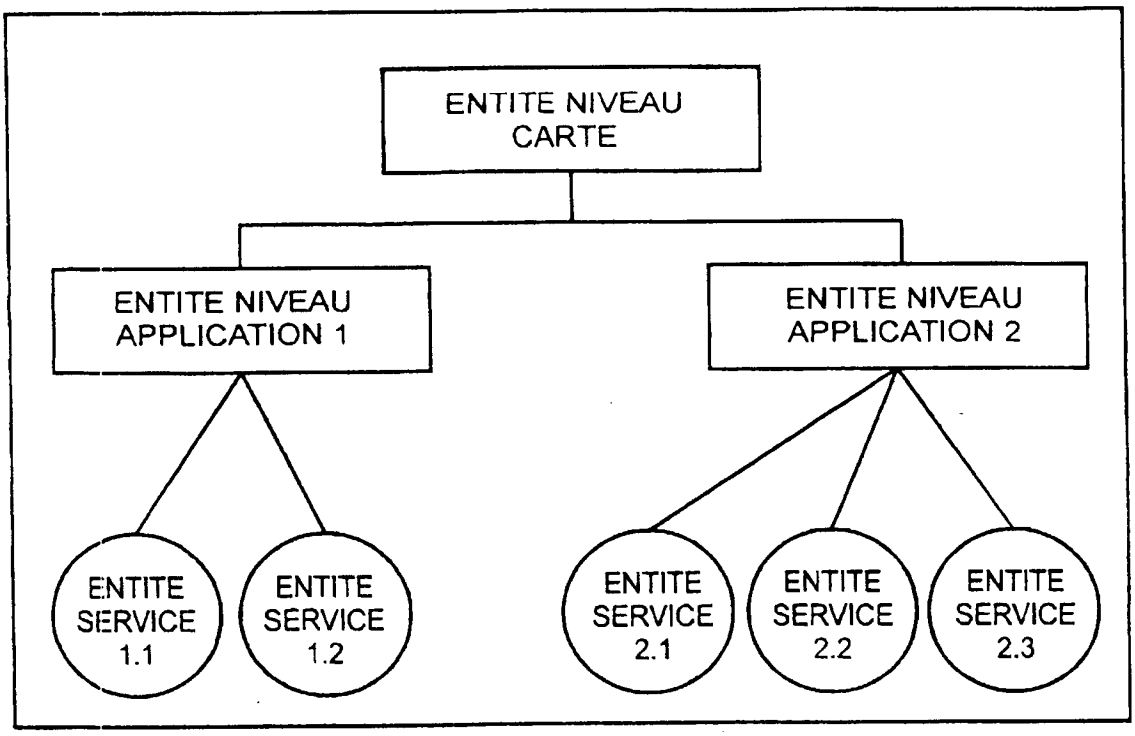


FIG. 1

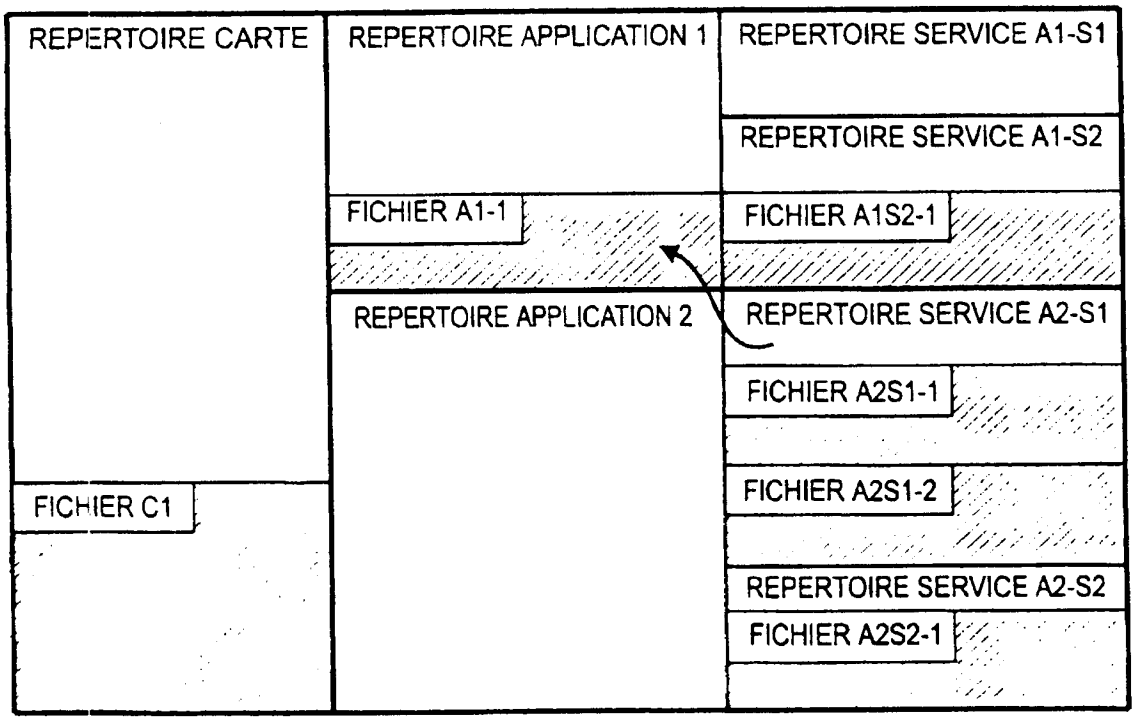


FIG. 2

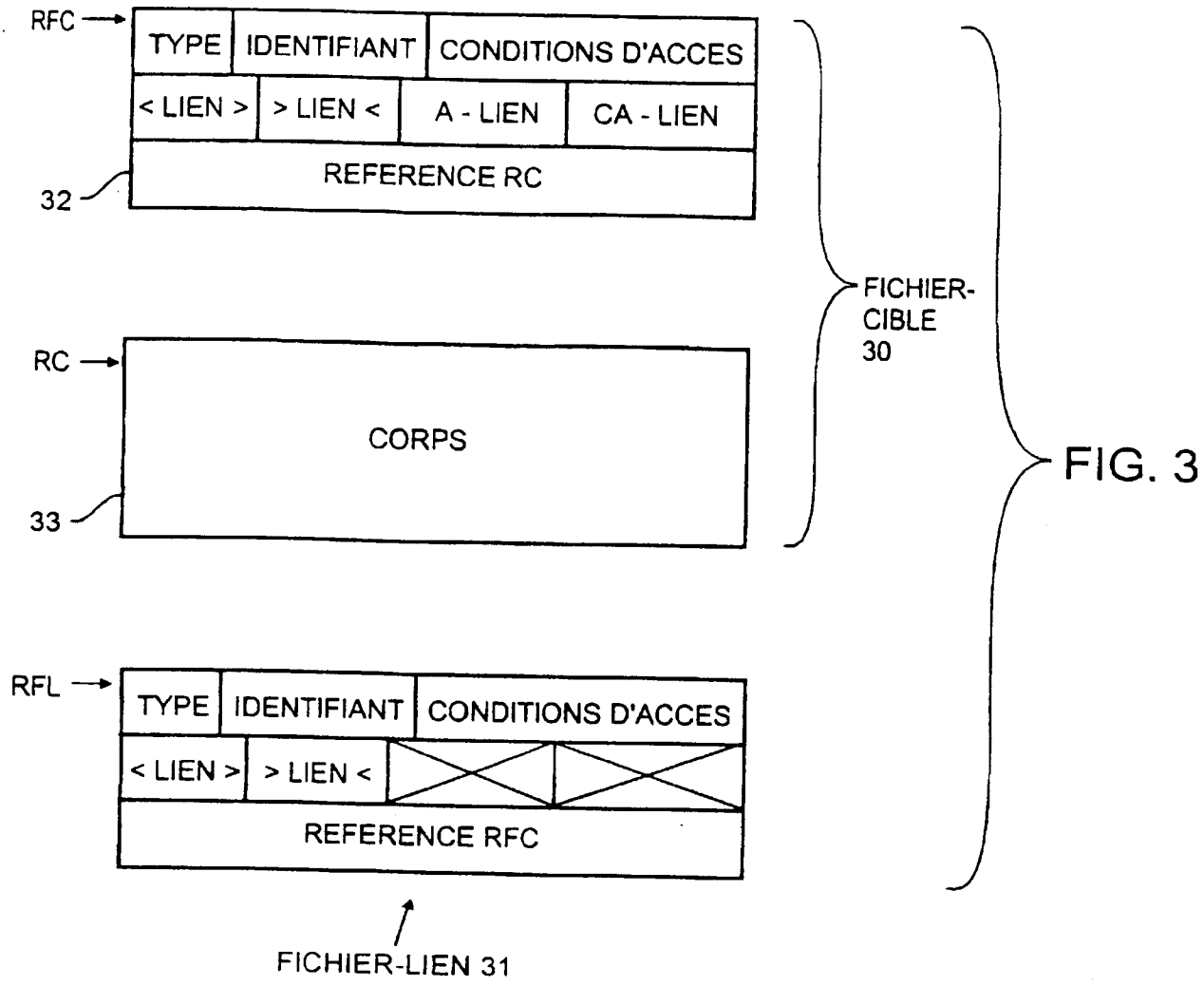


FIG. 3

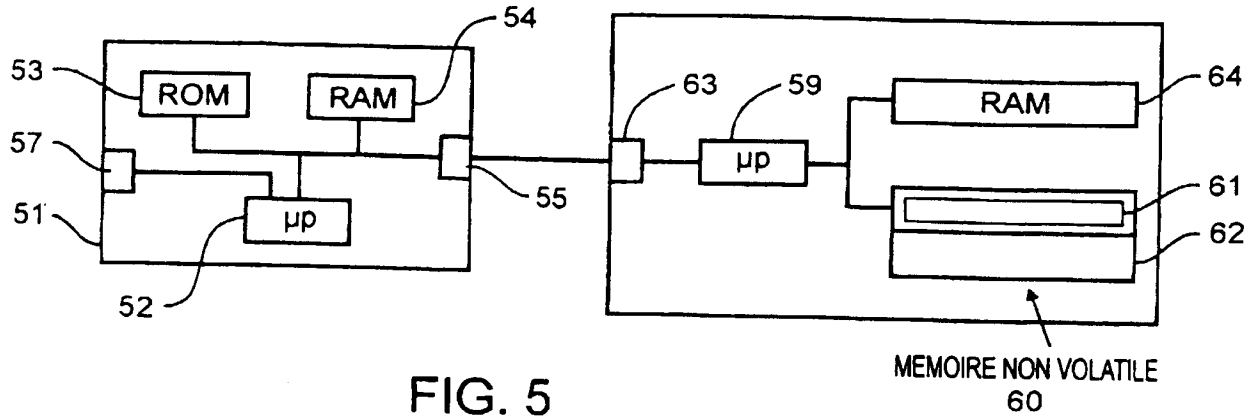


FIG. 5

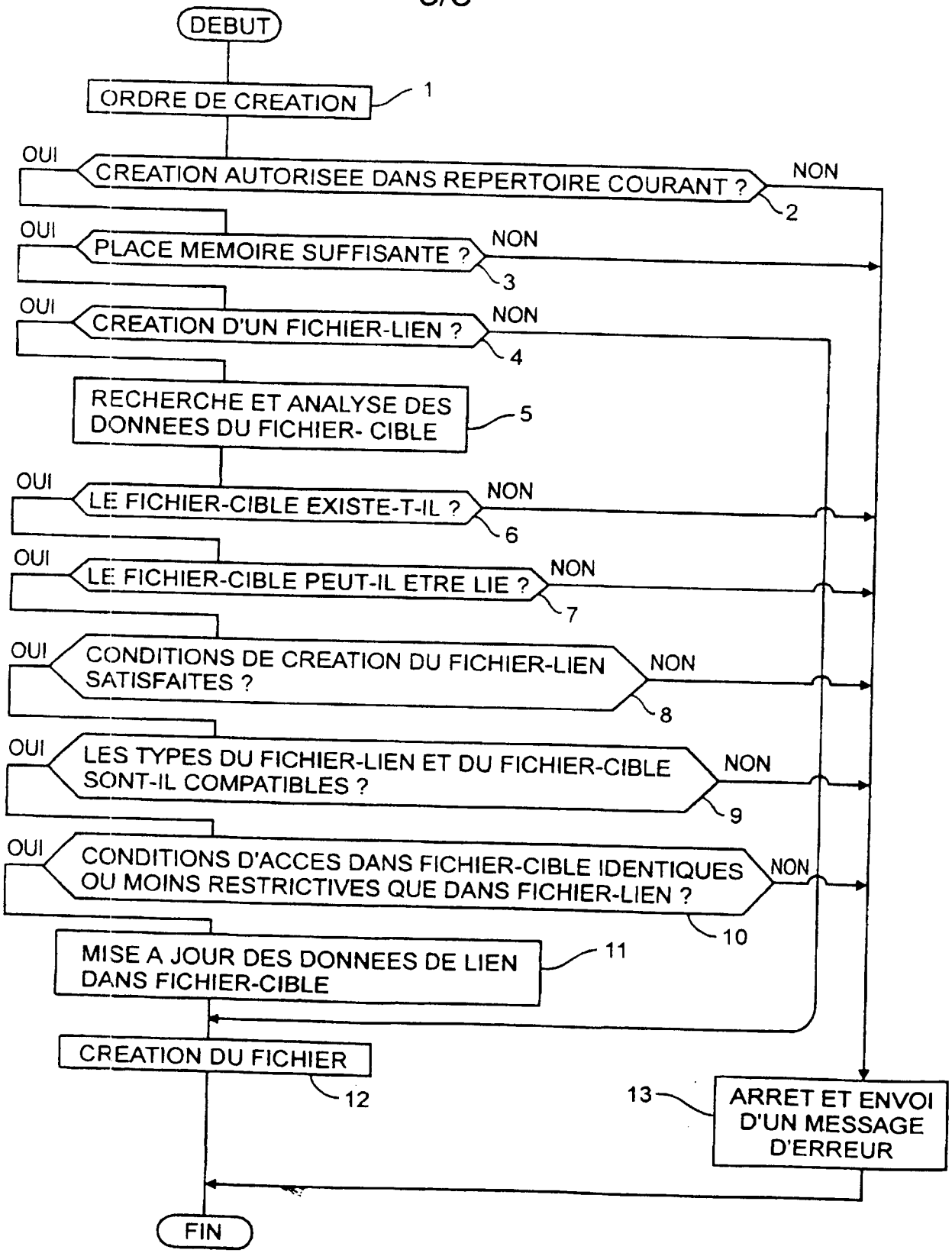


FIG. 4

REPUBLIQUE FRANÇAISE

INSTITUT NATIONAL
de la
PROPRIETE INDUSTRIELLE

RAPPORT DE RECHERCHE PRELIMINAIRE

établi sur la base des dernières revendications déposées avant le commencement de la recherche

2765362

N° d'enregistrement national

FA 544157
FR 9707996

DOCUMENTS CONSIDERES COMME PERTINENTS

Catégorie	Citation du document avec indication, en cas de besoin, des parties pertinentes	Revendications concernées de la demande examinée
X	US 4 960 982 A (TAKAHIRA KENICHI) * abrégé; figures 2,3 * * colonne 1, ligne 65 - colonne 2, ligne 18 *	1,2
Y		3,5
Y	US 5 479 509 A (UGON MICHEL) * abrégé; figure 1 * * colonne 2, ligne 1 - ligne 41 * * colonne 3, ligne 8 - ligne 38 *	3,5
A	EP 0 666 550 A (JONG EDUARD KAREL DE) * le document en entier *	1-3,5
A	US 5 497 418 A (KUDELSKI ANDRE) * colonne 4, ligne 4 - colonne 5, ligne 23 *	1-3,5
A	EP 0 332 117 A (TOKYO SHIBAURA ELECTRIC CO)	
A	GB 2 295 909 A (FUJITSU LTD)	

DOMAINES TECHNIQUES RECHERCHES (Int.Cl.6)

G07F
G06F

3

Date d'achèvement de la recherche: 31 mars 1998

Examinateur: Powell, D

CATEGORIE DES DOCUMENTS CITES

X : particulièrement pertinent à lui seul
Y : particulièrement pertinent en combinaison avec un autre document de la même catégorie
A : pertinent à l'encontre d'au moins une revendication ou arrière-plan technologique général
O : divulgation non-écrite
P : document intercalaire

T : théorie ou principe à la base de l'invention
E : document de brevet bénéficiant d'une date antérieure à la date de dépôt et qui n'a été publié qu'à cette date de dépôt ou qu'à une date postérieure.
D : cité dans la demande
L : cité pour d'autres raisons

& : membre de la même famille, document correspondant

EPO FORM 1503 03.82 (P04C13)